

PANDA: a software tool for improved train dispatching with focus on passenger flows

Ralf Rückert¹  · Martin Lemnian¹ ·
Christoph Blendinger² · Steffen Rechner¹ ·
Matthias Müller-Hannemann¹

Accepted: 3 October 2016 / Published online: 3 November 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract We introduce the decision support tool PANDA (Passenger Aware Novel Dispatching Assistance). Our web-based tool is designed to provide train dispatchers with detailed real-time information about the current passenger flow and the multi-dimensional impact of waiting decisions in case of train delays. After presenting the algorithmic background and PANDA's main features, we show how it can be utilized in a typical use case scenario for train dispatchers. Besides its practical value for train dispatchers, the framework can be used to systematically study scientific questions. Exemplarily, we use our software to experimentally analyse the influence of waiting decisions on realistic passenger flows of Deutsche Bahn. In a first experiment, we evaluate PANDA's potential benefit for passengers. Our findings indicate that a remarkable reduction in total delay might be possible in comparison to current practice. In two additional experiments, we investigate the timing aspect of waiting decisions. Our observations suggest that the timing of waiting decisions is of crucial importance and that a carefully implemented early rerouting strategy has a significant potential to reduce resulting delays of passengers.

✉ Ralf Rückert
rueckert@informatik.uni-halle.de

Martin Lemnian
mlemnian@informatik.uni-halle.de

Christoph Blendinger
Christoph.Blendinger@deutschebahn.com

Steffen Rechner
rechner@informatik.uni-halle.de

Matthias Müller-Hannemann
muellerh@informatik.uni-halle.de

¹ Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, Von-Seckendorff-Platz 1, 06120 Halle (Saale), Germany

² DB Mobility Logistics AG, Hahnstraße 40, 60528 Frankfurt am Main, Germany

Keywords Delays · Decision support · Passenger flows · Rerouting of passengers · Train dispatching · Waiting decisions

1 Introduction

In a fairly complex train traffic network like that of Germany, train disruptions and delays are inevitable. As a consequence, passengers may miss some planned transfer which results in significant delays at their destinations, in considerable dissatisfaction, and ultimately in economic loss for the railway company. To approach this problem, large railway companies like Deutsche Bahn employ train dispatchers who monitor delays and manually decide which trains shall wait for delayed incoming trains in order to maintain connections for passengers. Such waiting decisions induce further delays, and so train delay management becomes a highly complex optimization problem.

Unfortunately, only very limited information about current passenger flows is available for dispatchers. In today's practice, dispatchers make isolated waiting decisions based on experience and local knowledge. The induced side-effects on future transfers are usually only vaguely known. This lack of information leads to the policy of resolving transfer conflicts relatively late, when information about the arrival time is reliable. As a drawback, many possibilities for reroutings of passengers are not taken into consideration. We here investigate to which extent an early detection of conflicting transfers and corresponding rerouting policies is beneficial, that is, may have the effect of reducing average delays. Our working hypothesis is that detailed information about passenger flows is the key to improved waiting decisions. In our setting, we make the assumption that travel routes of each passenger are available, an assumption which is realistic only to a limited extent. At the moment, the most realistic passenger flow available to us is based on statistically validated demand models. We expect, however, that trends like electronic ticketing will enable railway operators to work with high-quality passenger flows in the foreseeable future.

1.1 Contribution

Extending the passenger-oriented delay management as introduced by Berger et al. (2011), we present **PANDA**, a webtool supporting train dispatchers with real-time information about the passenger flow and the impact of waiting decisions. **PANDA**'s key benefits for dispatchers are:

- an overview of all transfers requiring attention from dispatchers,
- detailed views on travel routes of all passenger groups who have planned to use a certain transfer,
- information about the impact of waiting decisions gained from simulation,
- evaluation of estimated arrival delays at the final destinations of all affected passenger groups, and
- based on such an impact model on passengers, a qualified recommendation to wait or not to wait, and where applicable, how many minutes to wait.

PANDA provides a testbed for studying and analysing dispatching strategies. We analyse the potential benefit of PANDA by a simulation experiment. PANDA evaluates for each critical transfer a number of decision criteria. PANDA only provides a recommendation to wait (or not to wait), if the majority of these criteria suggest waiting (or not waiting, respectively). For a set of ten test days in 2015, we simulate the effect of PANDA's waiting recommendation on the network and compare the results with the status quo. For those cases where PANDA is able to give a clear recommendation, we find that the average total delay can be reduced by about 20% and the number of passengers which suffer from large delays (namely, of 120 min or more) could be reduced by up to 50% with the help of PANDA.

Our software can also be used to analyse dispatching strategies. We analyse the effect of early rerouting on the passenger flow. Continuing previous work (Lemnian et al. 2014), we show that early rerouting is beneficial in a significant number of cases in comparison to the conservative strategy.

1.2 Related work

Train delay management is a well-studied field. We refer to Schachtebeck and Schöbel (2010), Kanai et al. (2011), and Dollevoet et al. (2012) for some recent work. Early delay management approaches simply assume that passengers who miss a connection wait for the next connection on the same line (for example, Schöbel 2006). Integrated passenger rerouting has been studied by Dollevoet et al. (2012), Dollevoet and Huisman (2014), and Kanai et al. (2011). Most of these approaches use integer linear programming (ILP) models and consider *offline versions*, where all delays are known before the optimization process starts. The integration of passenger rerouting into an ILP formulation leads to a considerable blow up, making these formulations so large for train networks like that of Germany with millions of passengers that they cannot be handled by standard integer programming techniques. In particular, solving such problems in an online setting seems to be out of reach without significant progress in solution techniques. Dollevoet and Huisman (2014) compare slow exact ILP solutions with a couple of fast heuristics. They introduce an iterative ILP approach which comes close to the exact solution but is significantly faster. However, it is not known whether the iterative ILP approach scales well to large-scale networks. Klierer and Suhl (2011) evaluate several simple dispatching rules in an *online scenario*, but work with randomly generated delay scenarios and randomly generated passenger flows while we use observed delays and more realistic passenger flows. Moreover, they work only with a subfleet of interregional trains from the Frankfurt area. Bauer and Schöbel (2014) also consider various online strategies. They introduce a learning strategy based on simulations with many delay scenarios and report promising result with this strategy in experiments on artificial schedules and generated delay data.

The most important conceptual difference with our work, however, is that all mentioned previous papers do not explore the question when to decide. Corman and Meng (2015) point out in a recent review paper “while online static rescheduling has reached a wide degree of dissemination, much is still to be done with regard to the probabilistic nature of the railway traffic rescheduling problems; and how to best take uncertainty into account for future states.”

Another line of research considers larger disruptions where part of the infrastructure is temporarily unavailable. Passenger-oriented management strategies for major disruptions have recently been proposed by Kroon et al. (2015) and Veelenturf et al. (2014).

2 Passenger-aware dispatching

The current practice in dispatching is to apply standard waiting time rules of the railway operators, which provide a pre-defined threshold for each transfer. If the delay of the feeding train is smaller or equal than this threshold, then the implied dispatching decision is that the connecting train must wait. If the threshold is slightly exceeded, then dispatchers may decide to maintain a specific connection. Especially in these cases **PANDA** supports the dispatcher in their decision making process. To do so, **PANDA** needs an accurate representation of the current state of the railway network and the passenger flow.

To model the railway schedule and passenger flow we use an *event-activity network* $\mathcal{N} = (\mathcal{V}, \mathcal{A})$, which is a directed acyclic graph with vertex set \mathcal{V} and arc set \mathcal{A} . Each vertex represents an arrival or departure event of some train. Arcs between vertices model relations between events. We distinguish between *driving arcs*, modelling the driving of a specific train until its next stop, *waiting arcs*, modelling a train standing on a platform, and *transfer arcs*, modelling the possibility for passengers to switch between trains. For a more extensive description of the *event-activity network* see Berger et al. (2011) or Lemnian et al. (2014). In our model, the passenger flow is represented by directed paths in \mathcal{N} . Each passenger corresponds to a path between a departure event at its starting station and an arrival event at its destination station. Passengers with an identical path in the event-activity network are merged to a group of passengers. In case of a disruption all passengers who belong to the same group are treated in the same fashion. This approach reduces the computational effort for persistently updating the passenger flow.

The basic workflow of our system looks as follows (Fig. 1). Whenever the event time of some event changes by some disruption or delay, we get a message which is used to update the state of our network by propagating the information through the network. While doing so, we classify each transfer arc into one of the states

- *safe*, meaning that the transfer will probably be maintained;
- *uncertain*, meaning that the transfer is uncertain but likely to hold. In this case, the feeder train is slightly delayed but by applying standard waiting time rules the transfer will be maintained;
- *critical*, meaning that the transfer will not hold but can be maintained by waiting just a few minutes;
- *broken*, meaning that the transfer will definitely not be maintained unless an exceptional waiting decision is taken.

For more information about the classification scheme and the classifiers see Lemnian et al. (2014).

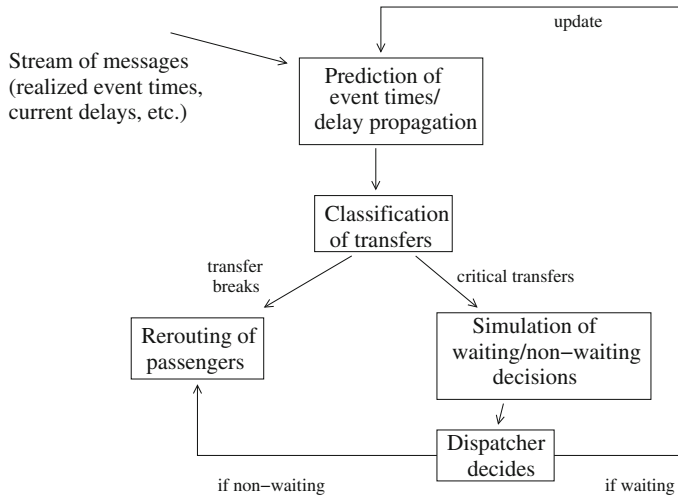


Fig. 1 Schematic sketch of the train dispatching process

Whenever a transfer is classified as *broken*, *critical*, or *uncertain*, it is displayed to the dispatcher by the PANDA webtool. In case of *critical* or *broken* transfers the dispatcher can simulate a waiting or non-waiting decision (Algorithm 1).

Algorithm 1 Simulate decision: waiting or non-waiting

Input: $\mathcal{N} := (\mathcal{V}, \mathcal{A})$, $t \in \mathcal{A}$ // event-activity network \mathcal{N} ; a critical or broken transfer t .

Output: $\mathcal{R} \in \mathbb{R}^{8 \times 2}$ // matrix that represents the 8 criteria for both decisions.

```

1:  $\mathcal{B} = \emptyset$  //  $\mathcal{B}$  denotes the set of affected passenger groups
2:  $d = \text{generateArtificialDelay}(t)$  //  $d$  is determined such that  $t$  is maintained
3:  $\mathcal{V}' = \text{propagateDelay}(d, \mathcal{N})$  // returns all delayed event nodes
4: for all  $v \in \mathcal{V}'$  do
5:    $\mathcal{B} = \mathcal{B} \cup \text{getPassengers}(v)$  // insert all affected passenger groups into  $\mathcal{B}$ 
6: end for
7:  $\mathcal{C} = \text{getConflicts}(\mathcal{V}')$  // returns all critical or broken transfers
8: for all  $t' \in \mathcal{C}$  do
9:   for all passenger groups  $p$  using  $t'$  do
10:     $\mathcal{B} = \mathcal{B} \cup \text{getPassengers}(p)$ 
11:     $\text{reroute}(p, \mathcal{N})$  // reroute passengers if necessary
12:   end for
13: end for
14:  $\mathcal{R} = 0_{8,2}$  // initialize evaluation criteria to zero
15:  $\text{computeResult}(\mathcal{B}, \mathcal{R}, 1)$  // evaluates decision criteria for waiting scenario (case 1)
16:  $\text{rollBack}(\mathcal{N}, \mathcal{B})$ 
17: for all passenger groups  $p$  using  $t$  do
18:    $\text{reroute}(p, \mathcal{N})$ 
19: end for
20: for all  $p \in \mathcal{B}$  do
21:    $\text{reroute}(p, \mathcal{N})$  // reroute passengers if necessary
22: end for
23:  $\text{computeResult}(\mathcal{B}, \mathcal{R}, 2)$  // evaluates decision criteria for non-waiting scenario (case 2)
24:  $\text{rollBack}(\mathcal{N}, \mathcal{B})$  // re-establishes original state before simulation
25: return  $\mathcal{R}$ 
  
```

If the dispatcher starts a simulation, PANDA first considers the waiting case. For the waiting case it is necessary to generate an artificial delay of the connecting train in such a way that the chosen transfer is maintained. This artificial delay is propagated in the same manner as a normal delay. During this propagation process all transfers are reclassified. The induced waiting conflicts are resolved by applying the standard waiting time rules used in Deutsche Bahn operations. Passenger groups that want to use transfers which are now classified as *critical* or *broken* need an alternative route. For all these groups PANDA calculates alternative routes with minimal arrival times. If more than one such route exists, we choose a route with a minimal number of transfers. The computed alternatives neglect available seat capacities and possible ticket constraints, since we currently do not possess such information. All groups that have been directly or indirectly affected by this waiting decision are stored in a set of *affected groups*. For a specific waiting decision, a passenger is directly affected if he/she wants to use the corresponding transfer or already sits in the departing train. In contrast, a passenger is only indirectly affected if, for example, a waiting decision causes a cascade of other waiting decisions and he/she sits in one of those trains which catch a secondary delay. After the passenger flow update, all passenger groups in this set will use feasible routes again. In the next step PANDA calculates the values of several evaluation criteria for the waiting decision based on all *affected groups*. The eight different criteria are explained in Sect. 3 (see *evaluation view*).

After the waiting decision is evaluated a roll back of all changes is performed to prepare the simulation of the non-waiting decision. In this simulation no artificial delay is injected to maintain the transfer. Instead PANDA calculates the best alternative route for each passenger group that wanted to use this transfer. After this step PANDA checks the feasibility of the current routes of all the remaining groups in the set of *affected groups*. All groups that use an infeasible route need to be rerouted. This rerouting step is necessary for two reasons: first, to leave the passenger flow in a consistent state and second, to make the results of both simulations comparable. Finally, PANDA calculates the values of the evaluation criteria for all *affected groups* of this alternative and displays the results to the dispatcher.

The time to execute a single simulation scales with the number of passenger groups for which alternative routes have to be calculated. Given state-of-the-art implementations of the corresponding routing queries, this can be done within a few seconds. Since these computations are independent and therefore can be done in parallel, they do not provide a bottleneck. This may change in the future when more sophisticated optimization techniques are integrated into the backend. The more time-consuming task for the dispatcher is to analyse and to interpret the complex result data which PANDA provides.

3 PANDA: a webtool supporting train dispatchers

PANDA is a web-based visualization software for realtime passenger flow information. It consists of a C++ backend, which models the event-activity network and performs all simulations, and a frontend for displaying the information

to the dispatcher which is based on the web programming framework *Ruby on Rails*. Next, we will focus on the description of the web-frontend.

3.1 Main view

The *main view* (Fig. 2) is intended to give the dispatcher a general overview of the current delay situation in the whole network. A map shows the stations where uncertain, critical or broken transfers will take place within the next minutes. To the right, a time-ordered list of these transfers is shown. A click on some transfer leads to the *transfer view* of the station where the transfer takes place.

3.2 Section view

A dispatcher who is responsible for a certain region may want to watch a smaller set of stations (in the same region or of some train line). To this end, the *section view* (Fig. 3) supports dispatchers by letting them observe a freely configurable list of selected stations. It provides an overview of all arrival and departure events as well as all uncertain, critical or broken transfers within the next few hours. Clicking on a specific transfer also leads to the *transfer view*.

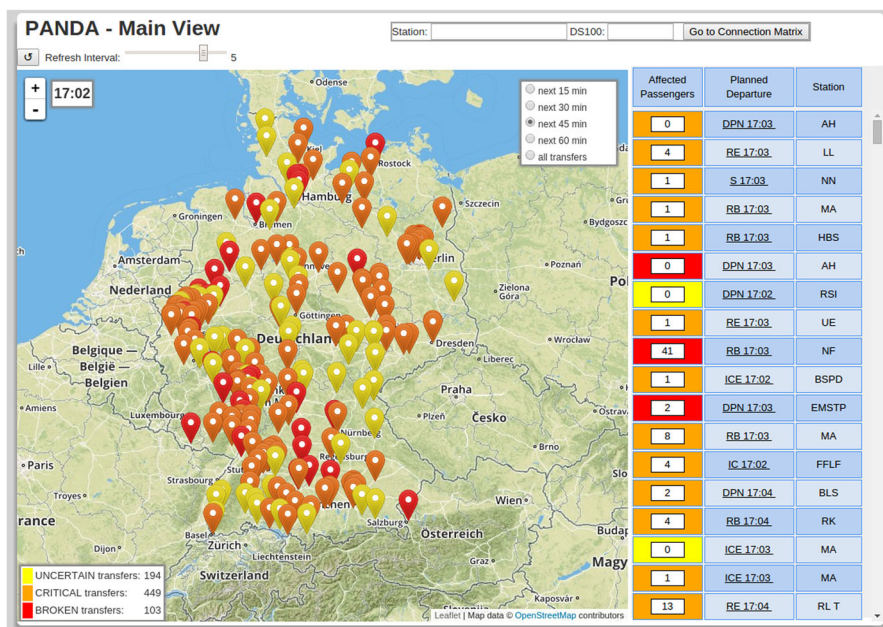


Fig. 2 Main view: uncertain (yellow), critical (orange), and broken (red) transfers are shown on the map and in an ordered list (color figure online)

3.3 Transfer view

The *transfer view* (Fig. 4) is divided into two major parts. On the left the dispatcher can see the station matrix. Each row (respective column) of this matrix corresponds to a departing (respective arriving) train at the selected station. When there is some planned transfer between trains, the relevant cell contains the number of passengers planning to use this transfer. A click on a cell selects a transfer for detailed information which is then shown on the right-hand side. This information contains the arrival and departure time of the feeder and connecting train, plus a listing of the affected passenger groups. The route of each group is shown as a coloured path on

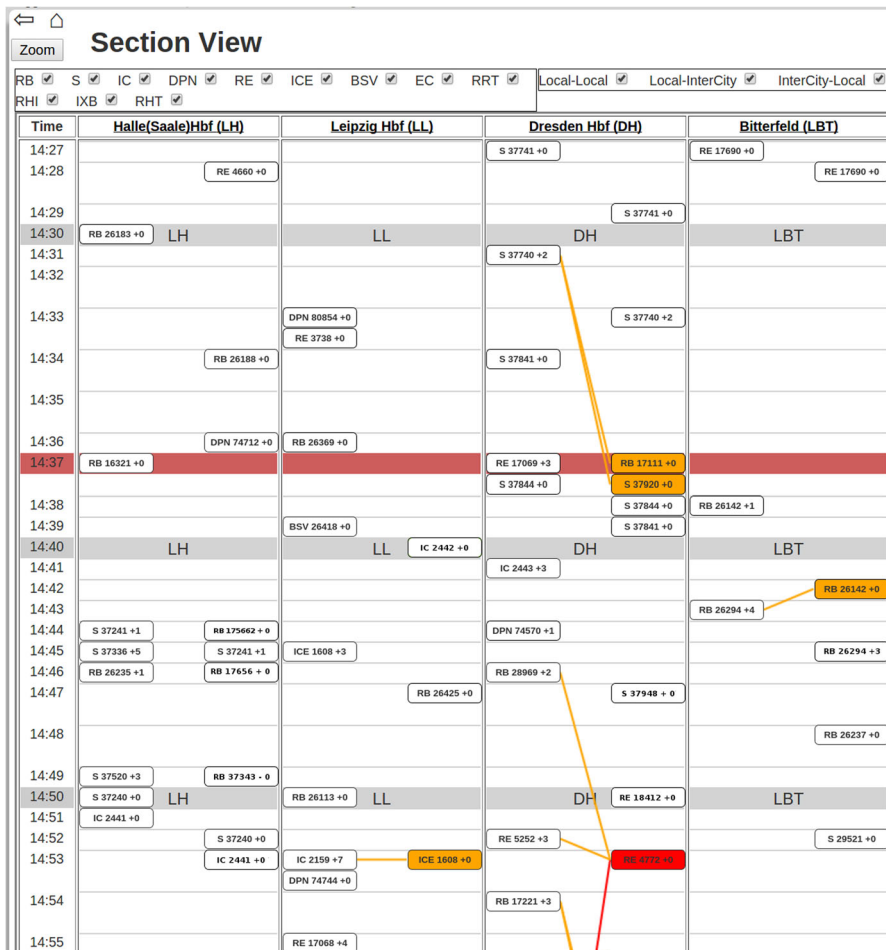


Fig. 3 Section view: each column shows arriving (left) and departing trains (right) at a certain station. Orange and red colors mark trains with critical transfers. The red horizontal bar displays the current time (02:37 p.m.) (color figure online)

the map below. Moreover, the dispatcher here has the option to simulate a waiting decision, meaning, he can simulate the connecting train to wait for transferring passengers and maintain the transfer, or not to wait. Both possibilities result in some delay of the affected passenger groups, which are compared in the *evaluation view*.

3.4 Evaluation view

This view (Fig. 5) shows the results of the simulation. Based on eight different criteria, both decisions are rated. The criteria contain the average delay of each passenger, as well as the number of passengers with small (≤ 5 min) and large (≥ 60 and ≥ 120 min) delay. Besides this valuable information, the *evaluation view* also shows the impact of both alternatives on the passenger flow. For example, waiting for transferring passengers results in additional delay for the connecting train. As an additional feature, the *evaluation view* also shows the routes of all passenger groups affected directly or indirectly by the decision.

4 A typical use case for dispatchers

We now describe how a dispatcher could use our software in a typical situation.

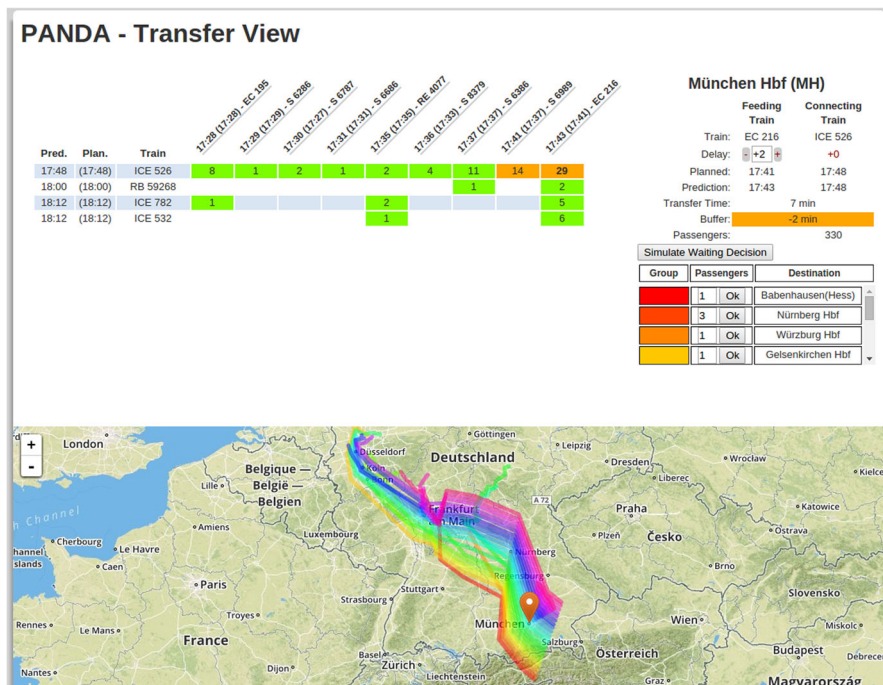


Fig. 4 Transfer view: station matrix and details of a selected transfer. Affected passenger groups are shown as *coloured paths* on the *map* (color figure online)

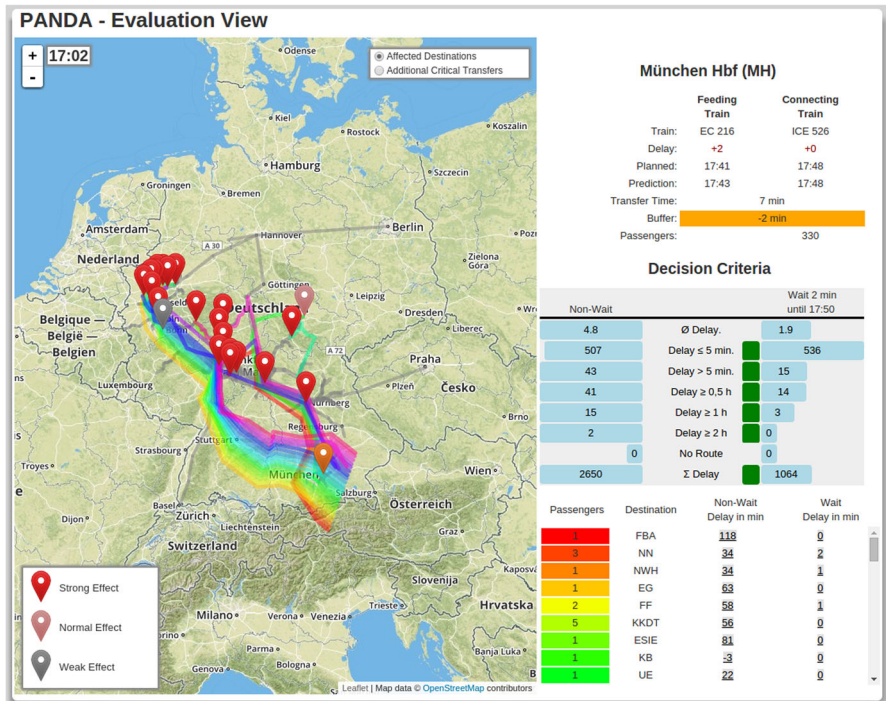


Fig. 5 Evaluation view: directly (coloured) and indirectly (grey) affected passengers are shown in the map. On the right, a summary statistic is displayed which allows to compare the effect of waiting and non-waiting decisions on passengers with respect to several criteria (color figure online)

1. It is 05:02 p.m. The dispatcher observes the *main view* (Fig. 2) and notices a critical delay (orange marker) at Munich Central Station (München Hbf) which he hasn't noticed before. The dispatcher wants to get additional information about this transfer and clicks on the corresponding marker.
2. The dispatcher is redirected to the *transfer view* (Fig. 4) where he gets information about the situation at Munich Central and in particular about critical transfers. He notices from the station matrix, that there are two critical (orange) transfers at this station within the next minutes. (S 6989 \Rightarrow ICE 526, and EC 216 \Rightarrow ICE 526, affecting 14, resp. 29 passengers). Further information about the latter transfer is also displayed. The dispatcher notices that the feeding train (EC 216) has a delay of 2 min. Due to a necessary transfer time of 7 min, the transfer is classified as critical. He has to decide whether the connecting train (ICE 526) should wait two additional minutes for transferring passengers. Such a decision would result in an additional delay of 2 min for 330 passengers, who are already sitting in ICE 526, in favour of 29 transferring passengers. To support the dispatcher in making a decision, PANDA offers the possibility to simulate both alternatives. The dispatcher clicks at the "Simulate Waiting Decision" button and, after a few seconds, is redirected to the *evaluation view*.

3. The results of the simulation are displayed in the *evaluation view* (Fig. 5). In this case, several criteria indicate that waiting for transferring passengers has a positive effect on the passenger flow. In particular, in the waiting scenario there are no passengers with a final delay of 2 h or more, while there are two passengers with such a large delay in the other case. In the displayed example, the evaluation criteria here clearly suggest to wait for transferring passengers. With the help of PANDA, the dispatcher can substantiate his passenger-friendly decision by arguments based on quantitative measures of several objective functions.

5 Case study: PANDA's potential benefit

To analyse the potential benefits of PANDA, we wanted to estimate the effect of PANDA's recommendations on the passenger flow. Therefore, we conducted an experiment in which we simulated and analysed several thousands of independent decisions. Here, we assume that all of PANDA's recommendations are realizable in practice. Thereby, we ignore problems of practical feasibility, for example due to track capacities.

5.1 Experimental setup

We use the German train schedule of 2015 (with about 66,000 trains and about 280,000 transfers used by passengers per day) and realistic passenger flow information provided by Deutsche Bahn, consisting of about 3.3 million passengers on 320,000 different routes on each day. For our evaluation we used recorded data for actual delays of 10 days in June and October 2015.

For each critical transfer on our test days, we simulated a waiting decision based on the delay situation as known 15 min before the connecting train was scheduled to depart. We measured the effect of each decision on the passenger flow and compare the results with the status quo. For the latter we used the realized event times. It may well happen that the delay situation changes in the last 15 min between the decision and its actual execution so that the decision would have been different with full knowledge of future delays.

Data about 154,000 critical cases were gathered this way. These cases also include transfers, which are not regarded in practice because of train-category priority rules. For instance, a dispatcher does not consider to delay a connecting subway train in order to keep the transfer from a delayed Intercity-Express train. After filtering out all such cases, an overall number of 106,000 critical transfers remained.

On each day, realized departure and arrival event times were available with a resolution of 1 min. However, we did not have explicit information whether a transfer has been maintained by an active dispatching decision or not. Therefore, we tried to infer a posteriori, based on the realized time stamps, for each critical transfer whether it had been dispatched or not. Given the actual departure time $dep(ct)$ of a connecting train ct , the actual arrival time $arr(ft)$ of a feeding train ft , and the

minimum transfer time *mintt* required to change from the feeding to the connecting train, the *buffer time* *buf* for a transfer can be calculated as $buf = dep(ct) - arr(ft) - mintt$. Since the recorded data available to us does not contain the information which transfers have been actually maintained, but we would like to compare the number of (actively) maintained connection in the status quo and by PANDA, a formal notion is required which can be computed from available data. Therefore, we counted a train as dispatched for a particular transfer if (1) it has departed at least 2 min later than possible (or in other words: if it has gathered a new delay at the departure event of at least 2 min) and the available buffer time satisfies $-1 \leq buf^1$ or (2) if its real departure time was at most 1 min earlier than calculated by PANDA for the waiting scenario.

To reduce potential misclassifications, we considered in our final evaluation only those transfers where—with respect to the time stamps 15 min before the scheduled departure of the connecting train—the departing train had to be delayed by at least 3 min to keep the transfer. After this filtering step, about 64,000 cases remained.

For each case, we asked PANDA about its waiting recommendations 15 min in advance. As PANDA offers several independent decision criteria, we used a simple majority rule to combine the criteria into a binary waiting recommendation: PANDA suggests to wait, whenever the number of criteria that are in favour of the waiting scenario exceeds the number of criteria that are in favour of the non-waiting scenario. To assess the impact on passengers, we consider four different measures, namely the total delay, i.e. the sum of the final delay over all passengers, and the number of passengers with a delay of at least 30, 60, or 120 min at their final destination.

5.2 Results

Under our assumptions, 28.2% of all critical connections are kept by dispatching. In comparison, PANDA recommends to wait for 31.8% of these cases. Table 1 gives the average absolute numbers of passengers per day which benefit from PANDA's recommendations, while Fig. 6 shows the relative benefit of using PANDA's recommendations in comparison with what has been executed in practice. For all measures, we observe a significant potential to reduce passenger delays by using recommendations given by PANDA. The strongest relative reduction can be obtained for the group of passengers which would suffer from delays by 120 or more minutes. The number of passengers in this group could be reduced by about 50%, if PANDA's recommendations would be realized (see again Table 1 for the absolute numbers of passengers affected per average day). The total sum of delay minutes could be reduced by about 24%. Recall, however, that not all recommendations are realizable due to other constraints.

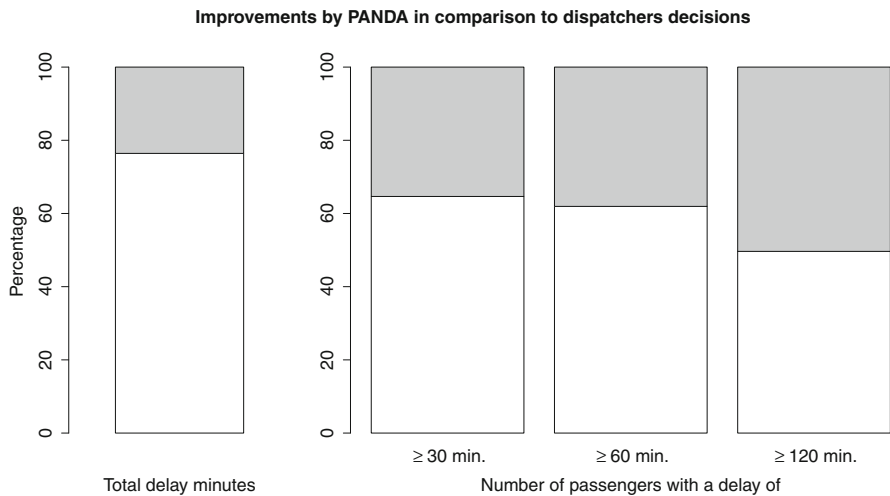
In Table 2, we report among all considered transfers the percentages of transfers maintained by PANDA, but not in the status quo, and conversely, by the status quo,

¹ Here, we relax the more natural condition of a non-negative buffer time slightly to reflect our observations in current practice. Dispatchers often calculate with a slightly shorter minimum transfer time than given in their “book of rules”.

Table 1 Comparison of PANDA's recommendation with status quo

x	Average # of passengers per day		
	Delay reduced to $< x$ min	Delay worsened to $\geq x$ min	Net benefit
30	9677.5	1610.2	8067.3
60	3879.4	717.8	3161.6
120	390.8	51.6	339.3

The table shows how many passengers would benefit from PANDA's recommendation, how many passengers would have an even worse solution, and the net benefit (difference of column 2 and 3) on average per day

**Fig. 6** Relative benefit of PANDA's recommendations. The *grey bars* indicate to what fraction the costs could be lowered when using PANDA's recommendations**Table 2** Comparison of PANDA's recommendation with status quo

		PANDA	
		WAIT (%)	NO-WAIT (%)
Status	WAIT	11.5	16.7
Quo	NO-WAIT	20.3	51.5

The table shows the percentage of cases among all waiting decisions, which fall into the four categories {status quo, WAIT}, {status quo, NO-WAIT}, {PANDA, WAIT}, {PANDA, NO-WAIT}

but not by PANDA. Likewise, we do this for the non-waiting decisions. We observe that there is a significant fraction of cases where the status quo and PANDA do not agree. In 16.7% of all cases status quo decides WAIT, but PANDA recommends

NO-WAIT, and conversely, in 20.3% of all cases PANDA decides WAIT, but status quo opts for NO-WAIT.

6 Case study: benefits of early rerouting

As mentioned before, the current policy of train dispatchers is to decide as late as possible. Typically a waiting decision is taken about 15 min (or even less) before a transfer takes place. The rationale behind this policy is that predictions about event times become more robust over time. Our working hypothesis is, however, that earlier detection of broken transfers and immediate rerouting leads to a smaller delay for most passengers, because more alternative routes are available. We conduct an experiment investigating this hypothesis, asking the question “What is the benefit of an early rerouting strategy for the passengers?”

We use the German train schedule of 2015 and realistic passenger flow information provided by Deutsche Bahn, consisting of about 3.3 million passenger routes. In our experiments we compare three strategies for rerouting: immediate rerouting after detection of some broken transfer, waiting with the rerouting until 15 min before the transfer would have taken place (which we call 15-min-in-advance strategy) and a hybrid strategy (Experiment C). Exemplarily, we display the results of the test day June 24, 2015.

6.1 Experiment A

We select the 1000 most used transfers on several test days in 2015. One after another, we artificially delay the feeder train by a large amount of time (for example, due to some malfunction of a train), enforcing a break of the chosen transfer and a rerouting of all those passengers who planned to use this transfer. These specific delays are assumed to be known 3 h in advance. This set-up allows us to compare immediate rerouting and the 15-min-in-advance strategy in isolation, that is, without any disturbing side effects resulting from complex changes in the delay scenario.

Figure 7 shows the results of Experiment A. Overall 22,681 passengers out of 3.3 million have been rerouted, and more than 28% of them have a benefit from immediate rerouting. As many as 5100 passengers (22.5%) have their delay reduced by at least 15 min, and 696 passengers (3.1%) by at least 1 h. These results show a significant potential for an immediate rerouting strategy under the assumption of full information about all future delays.

6.2 Experiment B

We use real-life delay information to evaluate our strategies. Note that this is a more realistic scenario than Experiment A. In this scenario, it is possible to worsen the situation of a passenger by immediate rerouting. For example, this happens if a passenger has been rerouted at an early stage of his trip but it has later turned out that no rerouting was necessary. That means, a planned transfer has been

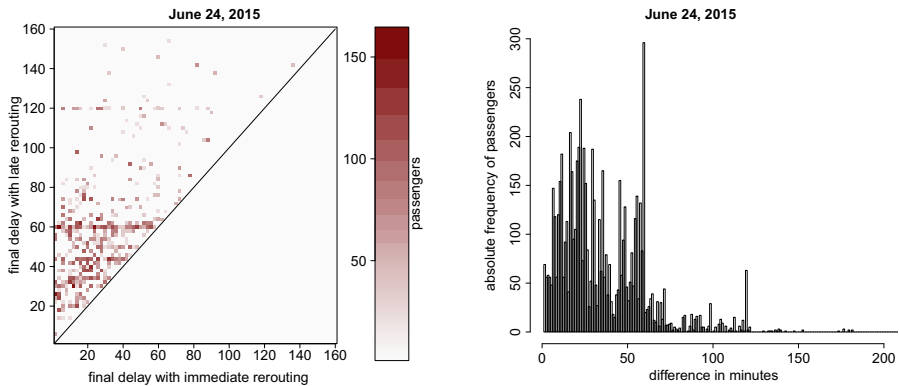


Fig. 7 Results of Experiment A: the heatmap (*left*) shows the final delay at the destination of all passengers for the two strategies: immediate rerouting (*x-axis*) vs. rerouting only 15 min in advance (*y-axis*). The histogram (*right*) shows the difference of the delays, i.e. the arrival time for the 15-min-in-advance strategy minus the arrival time of immediate rerouting

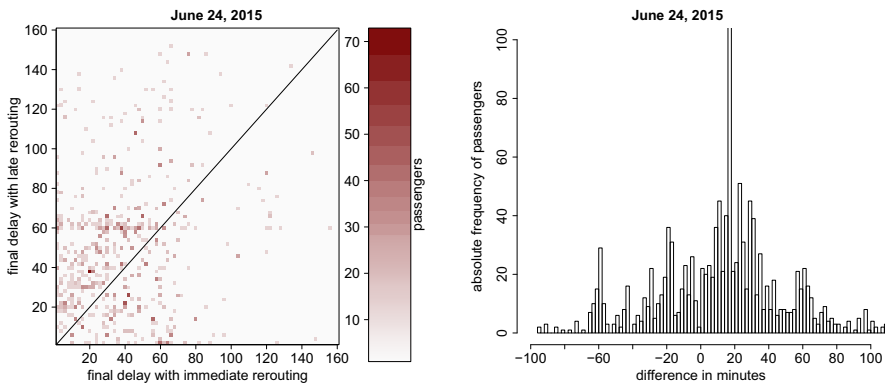


Fig. 8 Results of Experiment B: the heatmap (*left*) shows the final delay at the destination of all passengers for the two strategies: immediate rerouting (*x-axis*) vs. rerouting only 15 min in advance (*y-axis*). The histogram (*right*) shows the difference of the delays, i.e. the arrival time for the 15-min-in-advance strategy minus the arrival time of immediate rerouting. The cases with no difference are truncated from the heatmap and histogram for better visibility

misclassified as a broken transfer several hours in advance. Due to unexpected circumstances not only the feeder train but also the departing train may catch some delay, so that finally the transfer is maintained.

As expected, we observe a mixed picture for the real-life Experiment B where the delay scenario changes over time. Now both strategies have winners and losers. The immediate rerouting strategy produces more than twice as many winners as losers. The heatmap in Fig. 8 shows that many more cases lie above the diagonal than below. A point above the diagonal means that the final delay of the corresponding passenger is higher with the 15-min-in-advance strategy than with the immediate rerouting strategy. Likewise, we consider the difference of the arrival

times of both strategies (precisely, we take the arrival time of the 15-min-in-advance strategy minus the arrival time of the immediate rerouting strategy). Hence, if the difference is positive, this indicates an advantage for the immediate rerouting strategy. The right part of Fig. 8 shows the distribution of winners and losers. It has to be noted that for most rerouted passengers (namely for 88.4%) the strategy has no effect. Among the remaining passengers immediate rerouting has an advantage for 896 passengers (68.5%), whereas 411 passengers (31.5%) are better off with the 15-min-in-advance strategy. For those passengers where the choice of the strategy has an impact, the mean difference (that is, the mean reduction of the final delay) is about 14.7 min in favour of immediate rerouting. We conclude that the immediate rerouting strategy is beneficial from the perspective of an average passenger, but can also worsen matters for a non-negligible fraction of passengers.

6.2.1 Hybrid strategy: refined early rerouting

Motivated by the observed mixed picture in Experiment B we developed a third rerouting strategy. In that, we determine alternative routes as soon as we detect that a transfer might break. But in contrast to the immediate rerouting strategy, we wait with our recommendation as long as possible. More precisely, we compare the best available alternative with the currently used route and determine the first moment in time where the two routes differ. This gives us the latest moment where a passenger group could switch to that particular alternative. A few minutes before this moment we reconsider the current delay status. Since the first moment when we classified that the transfer might break, the delay situation may have changed. For example, the connecting train of the originally planned transfer also catches a delay so that our classification of it as being not maintained becomes obsolete. If we still predict the transfer to break, we perform a second search for an alternative route. We only reroute to the previously considered alternative route if the transfer is classified to break and if also the alternative is still feasible (it might be that meanwhile also the alternative route faces some previously unknown delay). By doing so, this refined strategy tends to avoid unnecessary reroutings of passengers.

6.3 Experiment C

We used the same experimental setup as in Experiment B, but this time we focus on the comparison of the immediate rerouting strategy with the hybrid strategy.

Again both strategies have winners and losers. The hybrid rerouting strategy produces more than twice as many winners as losers. The heatmap in Fig. 9 shows that more than twice as many cases lie above the diagonal than below. A point above the diagonal means that the final delay of the corresponding passenger is higher with the immediate rerouting strategy than with the new rerouting strategy. Likewise, we consider the difference of the arrival times of both strategies (precisely, we take the arrival time of the immediate rerouting strategy minus the arrival time of the new hybrid strategy). Hence, if the difference is positive, then this indicates an advantage for the hybrid strategy. The right part of Fig. 9 shows the distribution of winners and losers. For most rerouted passengers (namely for 98.5%)

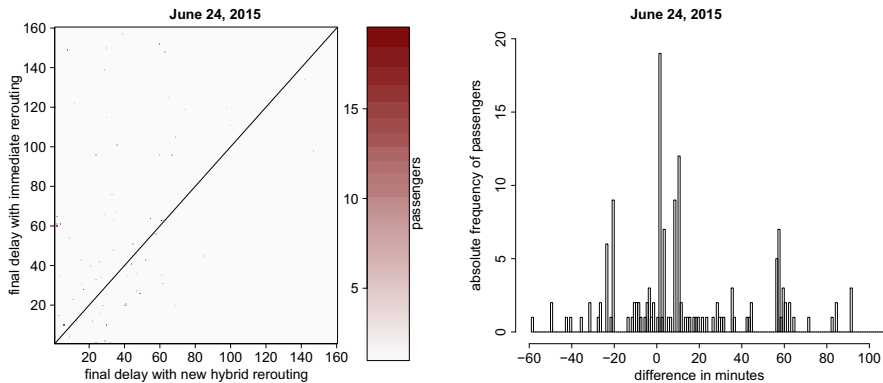


Fig. 9 Results of Experiment C: the heatmap (*left*) shows the final delay at the destination of all passengers for the two strategies: hybrid strategy (*x-axis*) vs. immediate rerouting (*y-axis*). The histogram (*right*) shows the difference of the delays, i.e. the arrival time for the immediate rerouting strategy minus the arrival time of the hybrid strategy. The cases with a zero minutes difference are truncated from the heatmap and histogram for better visibility

both strategies do the same. Among the remaining passengers the new hybrid strategy has an advantage for 121 (72.9%), whereas 45 (27.1%) are better off with the immediate rerouting strategy. For those passengers where the choice of the strategy has an impact, the mean difference (that is, the mean reduction of the final delay) is about 29 min in favour of the new hybrid strategy. We conclude that the hybrid strategy should be preferred over immediate rerouting.

7 Conclusions and future work

We presented the PANDA train dispatching software, a webtool designed to support train dispatchers by visualizing detailed real-time information about the passenger flow and the impact of waiting decisions. We conclude from our experiments that PANDA has a high potential to reduce the delay of passengers. However, in practice, dispatching decisions can not always be realized due to the layout and occupancy of tracks or other operational constraints. Nevertheless, we feel confident that PANDA is a valuable help for dispatchers. Additionally, the software has already been successfully tested within a field test with dispatchers.

We used PANDA to experimentally investigate the question whether early (immediate or hybrid) rerouting has a positive effect on the delay of passengers and found, that this is true in many cases. On average, such a strategy is beneficial and the hybrid strategy performs best.

However, the experiment with real data also shows that a non-negligible number of passengers may suffer from a higher final delay. As the occurrence of delays and disruptions is unforeseeable, there is clearly no perfect strategy which is optimal for all individual passengers. Further research is required to reduce the number of cases where the timing of rerouting decisions later turns out to be suboptimal.

In future, we want to conduct further analysis of dispatching strategies to support dispatchers by timing suggestions. A major future topic is the full optimization of waiting decisions. At the moment, we just simulate single waiting-or-non-waiting decisions subject to applying standard waiting time rules for all other transfers.

Acknowledgements The authors acknowledge partial support by Grant MU 1482/7-1 within the DFG research group FOR 2083 *Integrated Planning in Public Transport* and by Deutsche Bahn.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Bauer R, Schöbel A (2014) Rules of thumb—practical online strategies for delay management. *Public Transp* 6:85–105
- Berger A, Blaar C, Gebhardt A, Müller-Hannemann M, Schnee M (2011) Passenger flow-oriented train disposition. In: Demetrescu C, Halldórsson MM (eds) *Proceedings of the 19th annual European symposium on algorithms (ESA)*, Lecture notes in computer science, vol 6942. Springer, pp 227–238
- Corman F, Meng L (2015) A review of online dynamic models and algorithms for railway traffic management. *IEEE Trans Intell Transport Syst* 16:1274–1283
- Dollevoet T, Huisman D (2014) Fast heuristics for delay management with passenger rerouting. *Public Transp* 6:67–84
- Dollevoet T, Huisman D, Schmidt M, Schöbel A (2012) Delay management with rerouting of passengers. *Transport Sci* 46(1):74–89
- Kanai S, Shiina K, Harada S, Tomii N (2011) An optimal delay management algorithm from passengers' viewpoints considering the whole railway network. *J Rail Transp Plan Manag* 1:25–37
- Kliwer N, Suhl L (2011) A note on the online nature of the railway delay management problem. *Networks* 57:28–37
- Kroon LG, Maróti G, Nielsen LK (2015) Rescheduling of railway rolling stock with dynamic passenger flows. *Transport Sci* 49:165–184
- Lemnian M, Rückert R, Rechner S, Blendinger C, Müller-Hannemann M (2014) Timing of train disposition: Towards early passenger rerouting in case of delays. In: Funke S, Mihalák M (eds) *14th workshop on algorithmic approaches for transportation modelling, optimization, and systems, ATMOS 2014, OASICS*, vol 42. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, pp 122–137
- Schachtebeck M, Schöbel A (2010) To wait or not to wait and who goes first? Delay management with priority decisions. *Transport Sci* 44(3):307–321
- Schöbel A (2006) *Customer-oriented optimization in public transportation*. Springer, Berlin
- Veelenturf LP, Kroon LG, Maróti G (2014) Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. In: *10th international conference of the practice and theory of automated timetabling (PATAT 2014)*, pp 11–34